Pure Mathematics 理论数学, 2023, 13(12), 3401-3411
Published Online December 2023 in Hans. https://www.hanspub.org/journal/pm
https://doi.org/10.12677/pm.2023.1312353

Hans汉斯

基于均值漂移的三支聚类算法

吕军豪，徐　丁，孙　波，杨　晶

江苏科技大学理学院，江苏　镇江

收稿日期：2023年11月7日；录用日期：2023年12月7日；发布日期：2023年12月18日

摘　要

本文结合基于均值漂移的聚类算法与三支决策理论，首先利用核函数对中心点至样本点的向量进行加权求和，定义了偏移向量，据此不断移动中心点的位置，使样本中心点在密度梯度方向移动至密度最大的区域。然后根据样本点对类簇的访问频率将数据分为非噪声点和噪声点数据，对非噪声点数据采取传统的二支聚类得到核心域，对噪声点数据采取三支聚类，通过比较样本点对不同类簇的访问频率将样本点划分到相应类簇的边界域。将聚类结果用核心域和边界域表示。通过UCI数据集上的实验结果，验证了本文提出的算法相对于传统聚类可以提高聚类准确度、聚类结构的类内紧密度和类间分离度。

关键词

三支聚类，均值漂移，偏移向量

Three-Way Clustering Algorithm Based on Mean Shift

Junhao Lv, Ding Xu, Bo Sun, Jing Yang

School of Science, Jiangsu University of Science and Technology, Zhenjiang Jiangsu

Received: Nov. 7th, 2023; accepted: Dec. 7th, 2023; published: Dec. 18th, 2023

Abstract

By combining the clustering algorithm based on mean shift with the theory of three-way decision theory, this paper defines the mean shift vector according to the vector from the center point to the sample points, so that the center point of the samples is moved in the direction of the density gradient to the region of the highest density. According to the access frequency of the sample points to the class clusters the data are divided into non-noise point and noise point data, the traditional two-way clustering is taken to obtain the core domain for the non-noise point data, and

文章引用: 吕军豪, 徐丁, 孙波, 杨晶. 基于均值漂移的三支聚类算法[J]. 理论数学, 2023, 13(12): 3401-3411.
DOI: 10.12677/pm.2023.1312353

the three-way clustering is taken for the noise point data, and the sample points are divided into the boundary domains of the corresponding class clusters by comparing the access frequency of the sample points to the different class clusters. The clustering results were expressed in terms of core and boundary domains. The experimental results on the UCI dataset verify the advantages of the proposed algorithm over traditional clustering algorithms, which can improve the clustering accuracy, the intra-class closeness of the clustering structure and the inter-class separation.

Keywords

Three-Way Clustering, Mean Drift, Offset Vector

1. 引言

聚类[1]就是将一组数据对象集划分成几个类簇，使得处于同一个簇中的对象相似度较高，而处于不同簇中的样本相似度低。聚类在生物信息学[2]、图像处理[3]、企业管理[4]等领域有着广泛应用。聚类是一种无监督学习[5]方法，即在聚类的过程中没有数据集的原始类簇标签的信息，聚类算法也可以根据样本间相似度来进行划分，其优点是无需预先确定类簇个数也能进行聚类。按照聚类原理与数据结构的特性，聚类算法包括：基于层次的聚类算法，如凝聚层次聚类算法[6]和 BIRCH 算法[7]等；基于密度的聚类算法，如 DBSCAN 算法[8]和 DENCLUE 算法[9]等；基于模型的聚类算法，如 GMM 算法[10]和高斯混合模型算法[11]等。

传统聚类算法如层次聚类算法、DBSCAN 算法均为硬聚类，即数据集上的元素要么属于一个类，要么不属于一个类。当决策不完全信息下的对象时，简单直接地将某个对象划分到相应类簇中，往往会导致聚类精度下降和决策风险提高。同时，硬聚类对噪声点和异常点往往较为敏感，对不符合正态分布、离散分布的数据集的聚类结果有着较高的误差。

Fukunage [12]在 1975 年提出均值漂移聚类算法，其核心思想是通过滑动窗口进行局部密度估计，使得数据点朝向密度最大的区域聚集，从而实现聚类。后来，Cheng [13]在此基础上定义了核函数与权重系数，通过对不同样本点对偏移值的贡献值，优化了聚类的性能指标。均值漂移聚类算法是一种基于密度的无监督学习算法，相较于传统聚类算法的优点是能够通过数据的特征自适应地进行聚类，无需事先确定聚类的数量，且适用于处理大规模数据集和复杂数据类型。均值漂移聚类算法尽管改善了传统聚类算法的部分缺点，但对处理不完全信息问题的决策风险仍然较高。

为了降低传统聚类方法带来的决策风险，学者们尝试将三支决策思想引入无监督学习聚类，构建了许多基于三支决策的聚类算法。Lingras [14]等构建了粗糙聚类模型，引入粗糙集的正域、负域和边界域来表示聚类结果。Jiang [15]提出了基于三支决策的密度聚类算法，通过将 DBSCAN 算法与三支决策方法结合，引入数学形态学中的腐蚀和膨胀思想[16]，进行三支聚类，将聚类结果用核心域和边界域表示。Li [17]等提出了基于样本稳定性的三支聚类算法，根据每个样本的稳定性与设定阈值的关系，将样本分为稳定样本与不稳定样本，进行三支聚类得到核心域和边界域。

本文将在均值漂移算法的基础上引入三支决策思想，提出了一种基于三支决策的均值漂移聚类算法，传统的均值漂移聚类算法是一种基于密度的二支决策聚类算法，而本文根据对象访问不同类簇的次数是

否满足给定的收敛阈值条件，分离出聚类的核心域和边界域，对满足阈值条件的样本点进行划分至相应的核心域中；对不满足阈值条件的样本点进行延迟决策，即根据样本点访问类簇的次数将其划分至相应的边界域中，进而较好地解决了不确定信息带来的风险问题。根据核心域的聚类结果计算聚类评价指标，可以提高聚类结果的整体性能。

2. 相关工作

2.1. 三支决策聚类

三支决策是二支决策的延伸和拓展，2010 年由 Yao [18]在决策粗糙集的研究基础上提出的决策理论，其核心思想是将研究对象分为正域、负域和边界域，使其能有效降低信息不充分时的决策风险。3 个域分别对应 3 种决策规则，正域对应的是接受规则；负域对应的是拒绝规则；边界域对应的是不承诺规则。

目前，三支决策理论得到不断地充实与发展，并在各个领域得到广泛应用。Yu [19]将三支决策思想结合聚类模型，据此提出了三支聚类框架，即用核心域和边界域来代表一个类簇；提出了基于 ε 邻域的三支决策聚类模型，并引入数学形态学中腐蚀和膨胀思想得到核心域与边界域；Wang [20]提出了基于密度敏感谱聚类的三支决策聚类模型，引入容差参数和扰动分析得到类簇的核心域和边界域；Xu [21]提出了基于人工蜂群的三支 *k*-means 聚类算法，构造蜜源的适应度函数解决了初始聚类中心敏感的问题。

传统的二支聚类结果通常用独立的集合域来表示，二支聚类的集合域对应的是三支聚类的核心域 $Co(C)$，集合域互不相交，保证了集合域中的元素一定且仅属于此类，二支聚类的结果可以表示为：

$$U=\{Co(C_1),Co(C_2),\cdots,Co(C_k)\}$$

与传统的二支聚类不同，三支聚类通常使用三个互不相交的集合 $Co(C)$、$Fr(C)$、$Tr(C)$ 来分别表示核心域、边界域和外域。其中核心域中的元素一定属于此类，边界域中的元素可能属于此类，也有可能不属于此类，外域中的元素一定不属于此类。任意聚类结果应满足以下 3 个条件：

(1) $Co(C_i)\neq\varnothing$；

(2) $\bigcup_{i=1}^{k}\left(Co(C_i)\cup Fr(C_i)\right)=U$；

(3) $Co(C_i)\cap Co(C_i)=\varnothing, i\neq j$。

条件(1)保证任意类簇不能为空；条件(2)保证数据集 U 中的任意元素要么属于某个类簇中的核心域，要么属于某个或多个类簇中的边界域；条件(3)保证任意不同类簇之间有明显的界限，即它们的核心域是没有交集的。

因此，成对的核心域和边界域可以用来代表类簇的聚类结果，即三支聚类的结果可以表示为：

$$U=\{Co(C_1),Fr(C_1),Co(C_2),Fr(C_2),\cdots,Co(C_k),Fr(C_k)\}$$

2.2. 均值漂移算法(Mean Shift, MS)

均值漂移算法是基于核密度估计[22]的无监督聚类算法，即不需要提前确定类簇的数量，使其更符合实际情况。Mean Shift 算法的核心思想是在 ε 邻域内利用中心点至样本点的向量用核函数进行加权求和，得到中心点 x 的偏移向量，类簇的样本中心点根据偏移向量不断地沿着密度梯度方向移动，当中心点 x 的偏移向量的模 *shift* 小于给定的收敛阈值 *eps* 时，中心点最终落在密度最大的区域。最后，比较样本点 x_i 对不同类簇的访问次数，将其划分至其类簇中，完成聚类。

定义 1 (ε 邻域)

以给定区域中某样本点 p 为中心，ε 为半径的区域所包含样本点的集合，称为 ε 邻域，即

$$N_\varepsilon(p)=\{q\in U \mid dist(p,q)\le\varepsilon\}$$

其中，U 为总数据集，q 为数据集中的某一点，$dist(p,q)$ 为 p 和 q 之间的欧式距离。

定义 2 (偏移向量)

给定数据集 U 和 d 维空间 R^d，对空间中的任一样本点 $x_i\in U$，偏移向量的基本形式定义为

$$M_h(x)=\frac{1}{k}\sum_{x_i\in S_h}(x_i-x)$$

其中，k 表示在 n 个样本点 x_i 中有 k 个点落在 S_h 区域。S_h 表示以 x 为球心，h 为半径的球内所有数据点的集合，x_i 表示数据集 U 中第 i 个数据点，即

$$S_h=\{y:(y-x)^{\mathrm{T}}(y-x)\le h^2\}$$

偏移向量决定了中心点在下一次迭代的移动移动方向，将当前中心点 x 更新为 x 加上偏移向量 $M_h(x)$，即

$$x'=x+M_h(x)$$

x' 为下一次迭代时样本中心点的坐标。

定义 3 (收敛阈值 *eps*)

在均值漂移的迭代过程中，当中心点的偏移距离 *shift*，即偏移向量的模小于收敛阈值 *eps* 时，算法认为系统达到了最优聚类结果，并停止迭代。因此，收敛阈值又被称为终止准则。收敛阈值的选取需要综合考虑实际情况和数据结构特性，一般来说，较小的收敛阈值会增加迭代次数并产生更精确的聚类结果，较大的收敛阈值会减少迭代次数并产生更粗糙的聚类结果。

定义 4 (访问频率矩阵 $A_{n\times k}$)

由于均值漂移聚类算法不能直接输出聚类的结果，因此需要定义访问频率矩阵 $A_{n\times k}$，并根据每个样本点对不同类簇的访问次数的大小，对每个样本点进行类簇的划分，$A_{n\times k}$ 的基本形式为

$$A_{n\times k}=\begin{bmatrix}a_{11} & \cdots & a_{1k}\\ \vdots & \ddots & \vdots\\ a_{n1} & \cdots & a_{nk}\end{bmatrix}$$

$A_{n\times k}$ 的行数表示数据点的个数为 n，列数表示最终聚类的类簇数量 k。需要注意的是，类簇数量 k 不是事先确定的，而是通过数据的特征而自适应选择的。a_{ij} 表示第 i 个样本点对类簇 C_j 的访问次数的大小，则每个样本所属类簇则是该行中访问次数最大的那一类。

均值漂移聚类算法的核心在于求样本中心点的偏移向量，并根据最终样本中心的移动位置得到聚类结果。在给定数据集 U 和 d 维空间 R^d 的情况下，其详细步骤为：

初始化参数：给定 ε 邻域半径和收敛阈值 *eps*，将所有样本点标记为 *unvisited*；

初始化中心点：在被标记为 *unvisited* 的样本点中随机选取一个点作为类簇 C 的样本中心点 x_0；

更新访问次数：在 x_0 的 ε 邻域范围内找出所有样本点 x_i，记作集合 M，将集合 M 中的点在类簇 C 上的访问次数加 1，将访问次数保存在访问频率矩阵 $A_{n\times k}$ 的相应位置，同时认为这些点属于类簇 C；

求得偏移向量 M_r：以样本中心点 x_0 为圆心，ε 为半径，计算欧几里得空间下中心点 x_0 至 ε 邻域内所有样本点的向量，将向量求和得到样本中心点 x_0 的偏移向量 M_r；

更新中心点：根据偏移向量 M_r 和漂移公式 $x_0' = x_0 + M_r$，样本中心点 x_0 进行移动，即 x_0 沿着 $\vec{M}_r$ 方向移动了 $\|M_r\|$ 的距离；

停止迭代：重复上述操作，当偏移向量的模 $\|M_r\|$ 小于给定的收敛阈值 *eps* 时，停止迭代，将此时的样本中心点 x_0 存入样本中心点矩阵；

确定类簇数量：将样本中心点矩阵中的元素，即不同类簇之间的中心点 $x_0, x_1, \cdots, x_k$ 进行两两作差，若差值的模 $\|x_i - x_j\|$ 小于收敛阈值 *eps*，则将相应的两个类簇归并为同一类；若差值的模 $\|x_i - x_j\|$ 大于或等于收敛阈值 *eps*，则认为 x_j 对应的类簇 C_j 为新的类簇，增加一类；

划分聚类结果：重复上述步骤直到所有样本点均被标记为 *visited*，根据访问频率矩阵中的数值，将每个样本点划分至其访问次数最多的一类中，完成聚类。

MSC 算法流程如下：(表 1)

Table 1. MSC algorithm flowchart
表 1. MSC 算法流程图

算法 1：MSC 算法

输入：数据集 $U = \{x_1, x_2, \cdots, x_n\}$。

输出：聚类结果 $\{C_1, C_2, \cdots, C_n\}$。

1：初始化参数：邻域半径 ε，收敛阈值 *eps*，标记所有对象为 unvisited
2：while 存在被标记为 unvisited 的对象：
随机选择一个被标记为 unvisited 的对象，作为中心点 x_0；
for each $x_i \in x_0$ 的 ε 邻域 do：
if x_0 的 ε 邻域内的数据点 x_i 与 x_0 的距离 $d(x_i, x_0) \le \varepsilon$：
$M = \{x_i | x_i \in c\}$；
end if;
for each $x' \in M$ do：
偏移向量：$M_r = \sum_{i=1}^{n} (\overrightarrow{x_0 - x_i})$；
更新中心点：$x_0 = x_0 + M_r$；
if $\|M_r\| < eps$：记下此时中心点 x_0，停止迭代；
else 重复上述步骤；
end if;
end for;
end for;
end while;
3：while 类簇 c 收敛
if 当前类簇 c 与其他类簇 c' 的中心距离 $d(c, c') < M_r$：
合并 c, c'，记为 c'；
else 把类簇 c 看作新的类别，类簇个数 $k+1$；
end if;
end while;
4：for each $x \in U$ do：
if 数据点 x_i 对类簇 C 的访问次数 $t_i = \max\{t_1, t_2, \cdots, t_k\}$；
$x_i \in C$；
end for;
5：输出：聚类结果 $\{C_1, C_2, \cdots, C_n\}$。

3. 基于三支决策的均值漂移聚类(Meanshift Clustering Based on Three-Way Decisionm TWMSC)

MSC 算法是一种二支聚类算法，其基本步骤是首先随机选择一个未被访问的数据点作为中心点，计算中心点到以 ε 为半径内数据点的向量之和作为偏移向量，据此不断迭代更新中心点的位置，使中心点在密度梯度方向上移动，最终聚集在密度最大的区域。重复以上步骤直到所有点均被访问且收敛阈值 *eps* 满足条件时跳出循环。最后，根据访问频率矩阵对样本点进行聚类的划分，将样本点划分至其访问次数最多的类簇中，且认为该点仅属于此类。当某个样本点对多个类簇的访问频率相近时，MSC 算法只会将其划分至被访问次数最多的那一类中，而忽略了该样本点可能也属于被访问次数较多的那一类或几类中。

由此可见，尽管 MSC 算法在传统聚类算法的基础上进行了改进，具有能够根据数据特征自适应确定类簇个数，并且不需要初始化聚类中心从而避免局部最优解等优点，但与传统聚类算法相同的是，对不充分信息的数据点仍有着较高的决策风险。本文提出了基于三支决策的均值漂移聚类算法，即 TWMSC 算法，在 TWMSC 算法中考虑噪声数据点以及不充分信息带来的决策影响，通过均值漂移聚类的过程中噪声数据点的寻找和访问频率矩阵 $A_{n\times k}$ 对二支聚类的结果进行优化。

定义 1 (噪声数据点)

根据访问频率矩阵 $A_{n\times k}$ 的定义，将每个数据点对每个类簇的访问次数作为矩阵元素存入访问频率矩阵 $A_{n\times k}$，得到

$$A_{n\times k}=\begin{bmatrix} a_{11} & \cdots & a_{1k} \\ a_{21} & \ddots & \vdots \\ a_{n1} & \cdots & a_{nk} \end{bmatrix}$$

其中，第 i 个数据点对类簇 C_j 的访问次数记为 a_{ij}。设数据点 x_i 对类簇 $C_1,C_2,\cdots,C_k$ 的访问次数为 $a_{i1},a_{i2},\cdots,a_{ik},a_{ij}\in B$，记 $a_{i\max}=\max\{a_{i1},a_{i2},\cdots,a_{ik}\}$，若集合 B 中存在与 $a_{i\max}$ 数值相近的元素，即存在一个或多个类簇的被访问次数与访问所属类簇的次数相近，则认为该点为噪声数据点。

聚类的结果是通过寻找数据点 x_i 对所有类簇的访问次数 a_{ij} 最大的类簇，将其归属于类簇 C_i。在聚类的过程中，通常会出现以下两种情形：

1) $\mathrm{I}=\left\{s\middle|s=\dfrac{a_{ij}}{a_{i\max}}<\alpha,a_i=\max\{a_{i1},a_{i2},\cdots,a_{ik}\},a_{ij}\in B\right\}$；

2) $\mathrm{II}=\left\{s\middle|s=\dfrac{a_{ij}}{a_{i\max}}\geq\alpha,a_i=\max\{a_{i1},a_{i2},\cdots,a_{ik}\},a_{ij}\in B\right\}$。

其中，s 为访问次数的比例系数，$s\in(0,1)$，s 越大，说明该类簇的被访问次数越接近所属类簇被访问的次数；反之，则说明该类簇的被访问次数越小。α 为给定的比例阈值，$\alpha\in(0,1)$，用于判别数据点 x_i 能否被认为是噪声数据点，α 越大，则数据点 x_i 为噪声数据点的概率越小，更可能被划分至相应的核心域中；反之，α 越小，则数据点 x_i 为噪声数据点概率越大，更可能被划分至相应的边界域中。

若 x_i 属于情形 I，其访问次数 t_i 在该数据点对所有类簇的访问次数最大，且满足对其他类簇的访问次数与对类簇 C_i 的访问次数的比值 $<\alpha$，说明数据点 x_i 仅属于一个类簇 C_i 的核心域；若 x_i 属于情形 II，虽然其访问次数 t_i 在该数据点对所有类簇的访问次数最大，但存在该点对其他类簇的访问次数与对类簇 C_i 的访问次数的比值 $\geq\alpha$，可以认为该点是噪声数据点，应把该点看作满足访问次数要求的类簇的边界域。

由于 TWMSC 算法在得到访问频率矩阵 $A_{n\times k}$ 及之前的步骤与 MSC 算法完全一致，因此在给定数据集 U 和 d 维空间 R^d 的情况下，TWMSC 算法的具体步骤可以简化为：

1) 初始化参数：给定 ε 邻域半径和收敛阈值 eps；

2) 利用 MSC 算法得到聚类的访问频率矩阵 $A_{n\times k}$；

3) 根据数据特征确定比例阈值 α，并利用访问频率矩阵 $A_{n\times k}$ 计算比例系数 s，根据数据点 x_i 所属的情形，将数据点 x_i 分为噪声数据点与非噪声数据点；

4) 若 x_i 属于情形 I，则认为 x_i 是非噪声数据点，对 x_i 使用 MSC 算法，将其划分至被访问次数最多的类簇的核心域，进行直接决策；若 x_i 属于情形 II，则认为 x_i 是噪声数据点，对 x_i 使用 TWMSC 算法，记满足情形 II 条件的类簇为集合 P，将该数据点划分至集合 P 中每个类簇的边界域，进行延迟决策。

5) 完成聚类，将聚类结果用成对的核心域和边界域表示：

$$\left\{Co\left(C_1\right),Fr\left(C_1\right),Co\left(C_2\right),Fr\left(C_2\right),\cdots,Co\left(C_k\right),Fr\left(C_k\right)\right\}$$

TWMSC 算法流程如下：(表 2)

Table 2. TWMSC algorithm flowchart
表 2. TWMSC 算法流程图

算法 2：TWMSC 算法

输入：数据集 $U=\left\{x_1,x_2,\cdots,x_n\right\}$。

输出：聚类结果 $\left\{Co\left(C_1\right),Fr\left(C_1\right),Co\left(C_2\right),Fr\left(C_2\right),\cdots,Co\left(C_k\right),Fr\left(C_k\right)\right\}$。

1：初始化参数：邻域半径 ε，收敛阈值 eps，标记所有对象为 unvisited

2：while 存在被标记为 unvisited 的对象：

随机选择一个被标记为 unvisited 的对象，作为中心点 x_0；

for each $x_i\in x_0$ 的 ε 邻域 do：

if x_0 的 ε 邻域内的数据点 x_i 与 x_0 的距离 $d\left(x_i,x_0\right)\le\varepsilon$：

$M=\left\{x_i\middle|x_i\in c\right\}$；

for each $x'\in M$ do：

偏移向量：$M_r=\sum_{i=1}^{n}\left(\overrightarrow{x_0-x_i}\right)$；

更新中心点：$x_0=x_0+M_r$；

if $\left\|M_r\right\|<eps$：记下此时中心点 x_0，停止迭代；

else 重复上述步骤；

end if；

end for；

end for；

end while；

3：while 类簇 c 收敛

if 当前类簇 c 与其他类簇 c' 的中心距离 $d\left(c,c'\right)<M_r$：

合并 c,c'，记为 c'；

else 把类簇 c 看作新的类别，类簇个数 $k+1$；

end if；

end while；

4：for each $x_i\in U$ do：

if 数据点 x_i 对类簇 C 的访问次数 $t_i=\max\left\{t_1,t_2,\cdots,t_k\right\}$；

for each $t_j, j\ne i$ do：

if $\forall\frac{t_j}{t_i}<\alpha, j=1,2,\cdots,k$：

Continued

$x_i \in Co(C_i)$；

if $\exists \frac{t_j}{t_i} \geq \alpha, j = 1,2,\cdots,k$：

$x_i \in Fr(C_i), x_i \in Fr(C_j)$；

end if；

end for；

end if；

end for；

5：输出：聚类结果

$\{Co(C_1), Fr(C_1), Co(C_2), Fr(C_2), \cdots, Co(C_k), Fr(C_k)\}$。

4. 实验结果

为了验证本文算法的高效性，本文选取 6 组 UCI 数据集对算法进行验证。UCI 数据集的数据量适中、数据质量较高、数据噪声点少等优点，因此被广泛认可。数据集如表 3 所示。

Table 3. The dataset used in the experiment
表 3. 实验中使用的数据集

数据集	样本个数	样本维度	类簇数
seeds	210	7	3
ecoli	336	7	8
D31	3100	2	31
iris	150	4	3
S2	5000	2	15
R15	600	2	15

聚类的评价指标大多是通过紧凑性和可分性来定义的：紧凑性是衡量聚类中元素彼此之间的距离，可分性是衡量不同类簇之间的距离。聚类的评价指标可分为两类：外部指标和内部指标。外部指标对应的是外部评估方法，是指在知道真实标签的情况下来衡量聚类结果的好坏，例如准确率[23] (ACC)，兰德指数[24] (Rand Index)等。内部指标对应的是内部评估方法，是指仅通过数据来衡量聚类结果的好坏，常见的有平均轮廓系数[25] (AS)，Davies-Bouldin 指数[26] (DBI)等。本文所选用的评价指标为准确率(Accuracy)，平均轮廓系数(Average Sihouette Coefficient)和 DBI，其中，准确率和平均轮廓系数值越高，聚类效果就越好，DBI 值越小，聚类效果越好。

为了比较基于均值漂移的三支聚类算法与传统聚类算法的有效性，本文选取传统的 *k*-means 聚类算法和 MSC 算法对选取的 6 组 UCI 数据集进行二支聚类。同时根据经验调整邻域半径 ε 和收敛阈值 *eps*，使用 TWMSC 算法，对选取的 6 组 UCI 数据集进行聚类，再选取聚类评价指标 ACC、AS 和 DBI 对三种聚类结果进行评价。本实验对选取的每组数据集进行 50 次重复实验，聚类评价指标性能用均值代替，实验结果如表 4 所示。

由实验结果得知：TWMSC 算法能够有效地改善聚类评价指标 ACC、AS 和 DBI，在某些数据集上的聚类结果明显优于传统的 *k*-means 算法和 MSC 算法，聚类效果的总体性能有着一定的提升。以 iris 数据集为例，本文给出的 TWMSC 算法相较于 *k*-means 算法和 MSC 算法，ACC 和 AS 变大，DBI 变小，同时提高了聚类准确度、聚类结构的类内紧密性和类间分离性。即使在某些数据集上 TWMSC 算法的聚类

Table 4. The clustering results of the dataset used in the experiment
表 4. 实验中使用的数据集的聚类结果

数据集	算法	ACC	AS	DBI
seeds	*k*-means	0.8547	0.4681	0.8757
	MSC	0.8286	0.4693	0.8182
	TWMSC	**0.8952**	**0.4719**	**0.788**
ecoli	*k*-means	0.5685	0.2289	0.5397
	MSC	0.5863	0.2149	0.5112
	TWMSC	**0.9309**	**0.2575**	**0.505**
D31	*k*-means	0.9087	0.5317	**0.8969**
	MSC	0.9577	0.4993	0.9491
	TWMSC	**0.9609**	**0.533**	0.9441
iris	*k*-means	0.8867	0.551	0.8701
	MSC	0.9333	0.5363	0.8689
	TWMSC	**0.9362**	**0.5526**	**0.5827**
S2	*k*-means	0.8934	0.5668	0.8957
	MSC	0.8688	0.5519	0.8734
	TWMSC	**0.9175**	**0.6262**	**0.8653**
R15	*k*-means	0.9678	**0.6871**	0.9189
	MSC	0.97	0.6719	0.8753
	TWMSC	**0.9717**	0.6748	**0.8055**

效果不如传统的 *k*-means 算法，例如 R15 数据集上的 AS，但 DBI 仅下降了 0.18%，对聚类效果的总体性能影响不大。相较于传统聚类算法，TWMSC 算法无需预先确定聚类的数量以及初始样本中心点，能够根据实际的数据特征自适应调整类簇数量，并且可以精确高效地定位样本点的密集区域，因此聚类的准确度得到提高。此外，三支决策思想的引入有效地解决了非离散分布和非正态分布的数据集等不充分信息下的问题。

综上所述，本文提出的 TWMSC 算法能够有效提高聚类结果的 ACC、AS 等性能指标，相较于传统的聚类算法和 MSC 算法有着一定的优势，可以说明基于均值漂移的三支聚类算法是高效可行的。

5. 结束语

本文提出的基于均值漂移的三支聚类算法在预先不确定聚类数量的情况下为处理不完整信息的聚类问题提供了一种有效的解决方案。在处理高维数据和大规模数据时，该算法能快速准确地定位密集区域，提高了聚类的总体性能，在数据分析、模式识别和数据挖掘等领域具有指导意义。考虑到 TWMSC 算法中的参数是根据实际情况人工选取的，因此在聚类的过程中会存在一定偏差。因此，在本文研究的基础上，未来的研究可以进一步探索：1) 如何根据数据结构特征自适应地选取收敛阈值 *eps* 等参数，进而提升算法的可靠性。2) 如何降低离群点对参数的自适应选取以及聚类结果的影响，使得算法高效可行且不会陷入局部最优解。

基金项目

本文受到江苏省高等学校大学生创新创业训练计划项目资助。

参考文献

[1] Dong, W., Yusuke, N., Motohisa, S. and Aketagawa, M. (2021) Cluster Analysis Based Fringe-Activity Range Detector. *Optics Communications*, **483**, Article ID: 126626. https://doi.org/10.1016/j.optcom.2020.126626

[2] Rezaul, M.K., Oya, B., Achille, Z., *et al*. (2020) Deep Learning-Based Clustering Approaches for Bioinformatics. *Briefings in Bioinformatics*, **22**, 393-415. https://doi.org/10.1093/bib/bbz170

[3] Song, L. and Zhang, X. (2018) Improved Pixel Relevance Based on Mahalanobis Distance for Image Segmentation. *International Journal of Information and Computer Security*, **10**, 237-247. https://doi.org/10.1504/IJICS.2018.10012573

[4] Mykhailo, V., Maria, Z., Lesia, S.V., *et al*. (2020) Management of the Social Package Structure at Industrial Enterprises on the Basis of Cluster Analysis. *TEM Journal*, **9**, 249-260.

[5] Rai, P.K. and Dwivedi, K.R. (2012) Clustering Techniques for Unsupervised Learning. *International Journal of Managment, IT and Engineering*, **2**, 462-571.

[6] Liu, H., Fen, L., Jian, J., *et al*. (2018) Overlapping Community Discovery Algorithm Based on Hierarchical Agglomerative Clustering. *International Journal of Pattern Recognition and Artificial Intelligence*, **32**, Article ID: 1850008. https://doi.org/10.1142/S0218001418500088

[7] Fanny, R., Muhammad, Z. and Saib, S. (2020) Improve BIRCH Algorithm for Big Data Clustering. *IOP Conference Series*: *Materials Science and Engineering*, **725**, Article ID: 012090. https://doi.org/10.1088/1757-899X/725/1/012090

[8] Zuo, Y., Hu, Z., Yuan, S., *et al*. (2022) Identification of Convective and Stratiform Clouds Based on the Improved DBSCAN Clustering Algorithm. *Advances in Atmospheric Sciences*, **39**, 2203-2212. https://doi.org/10.1007/s00376-021-1223-7

[9] Rehioui, H., Idrissi, A., Abourezq, M. and Zegrari, F. (2016) DENCLUE-IM: A New Approach for Big Data Clustering. *Procedia Computer Science*, **83**, 560-567. https://doi.org/10.1016/j.procs.2016.04.265

[10] Shi, J., He, Q. and Wang, Z. (2019) GMM Clustering-Based Decision Trees Considering Fault Rate and Cluster Validity for Analog Circuit Fault Diagnosis. *IEEE Access*, **7**, 140637-140650. https://doi.org/10.1109/ACCESS.2019.2943380

[11] Shi, J., Liu, X., Yang, S., *et al*. (2021) An Initialization Friendly Gaussian Mixture Model Based Multi-Objective Clustering Method for SAR Images Change Detection. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-020-02584-w

[12] Fukunaga, K. and Hostetler, L. (1975) The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. *IEEE Transactions on Information Theory*, **21**, 32-40. https://doi.org/10.1109/TIT.1975.1055330

[13] Cheng, Y. and Fu, K.S. (1985) Conceptual Clustering in Knowledge Organization. IEEE Transactions on Pattern *Analysis and Machine Intelligence*, **PAMI-7**, 592-598. https://doi.org/10.1109/TPAMI.1985.4767706

[14] Lingras, P. and West, C. (2004) Interval Set Clustering of Web Users with Rough K-Means. *Journal of Intelligent Information Systems*, **23**, 5-16. https://doi.org/10.1023/B:JIIS.0000029668.88665.1a

[15] 姜凡. 基于三支决策的密度聚类算法[J]. 应用数学进展, 2022, 11(2): 858-865.

[16] Wang, P. and Yao, Y. (2018) CE3: A Three-Way Clustering Method Based on Mathematical Morphology. *Knowledge-Based Systems*, **155**, 54-65. https://doi.org/10.1016/j.knosys.2018.04.029

[17] 李刘万, 朱金, 王平心. 基于样本相似度的三支聚类算法[J]. 扬州大学学报(自然科学版), 2022, 25(6): 40-44.

[18] Yao, Y. (2010) The Superiority of Three-Way Decisions in Probabilistic Rough Set Models. *Information Sciences*, **181**, 1080-1096. https://doi.org/10.1016/j.ins.2010.11.019

[19] Yu, H., Zhang, C. and Wang, G. (2016) A Tree-Based Incremental Overlapping Clustering Method Using the Three-Way Decision Theory. *Knowledge-Based Systems*, **91**, 189-203. https://doi.org/10.1016/j.knosys.2015.05.028

[20] 凡嘉琛, 王平心, 杨习贝. 基于三支决策的密度敏感谱聚类[J]. 山东大学学报(理学版), 2023, 58(1): 59-66.

[21] 徐天杰, 王平心, 杨习贝. 基于人工蜂群的三支 k-means 聚类算法[J]. 计算机科学, 2023, 50(6): 116-121.

[22] Taaffe, K., Pearce, B. and Ritchie, G. (2021) Using Kernel Density Estimation to Model Surgical Procedure Duration. *International Transactions in Operational Research*, **28**, 401-418. https://doi.org/10.1111/itor.12561

[23] Wang, Z., Farhand, S. and Tsechpenakis, G. (2019) Fading Affect Bias: Improving the Trade-Off between Accuracy

and Efficiency in Feature Clustering. *Machine Vision and Applications*, **30**, 255-268. https://doi.org/10.1007/s00138-019-01008-w

[24] Warrens, W. and van der Hoef, H. (2022) Understanding the Adjusted Rand Index and Other Partition Comparison Indices Based on Counting Object Pairs. *Journal of Classification*, **39**, 487-509. https://doi.org/10.1007/s00357-022-09413-z

[25] Bagirov, A.M., Aliguliyev, R.M. and Sultanova, N. (2023) Finding Compact and Well-Separated Clusters: Clustering Using Silhouette Coefficients. *Pattern Recognition*, **135**, Article ID: 109144. https://doi.org/10.1016/j.patcog.2022.109144

[26] Davies, D.L. and Bouldin, D.W. (1979) A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **1**, 224-227. https://doi.org/10.1109/TPAMI.1979.4766909